ФГОБУ ВО «Финансовый университет при Правительстве Российской Федерации»
ВСЕРОССИЙСКАЯ ОЛИМПИАДА ПО ИНФОРМАТИКЕ
«МИССИЯ ВЫПОЛНИМА. ТВОЕ ПРИЗВАНИЕ – ФИНАНСИСТ!»
ОЧНЫЙ ЭТАП, 2023 год

Продолжительность олимпиады – 235 минут. Олимпиадное задание состоит из пяти задач. Для каждой задачи указан ее вес в баллах.
Участник олимпиады самостоятельно определяет последовательность выполнения задач. На одном из языков программирования – C/C++, C#, Visual Basic, Pascal или Python – разработайте *консольные* программы для решения перечисленных ниже задач.

При выполнении задания участник формирует каталог в имени которого указывает свое ФИО. В данном каталоге формирует пять каталогов: Task1; Task2; Task3; Task4; Task5. Решение задачи размещаются в каталоге с соответствующим номером (см. рис П.1**)**

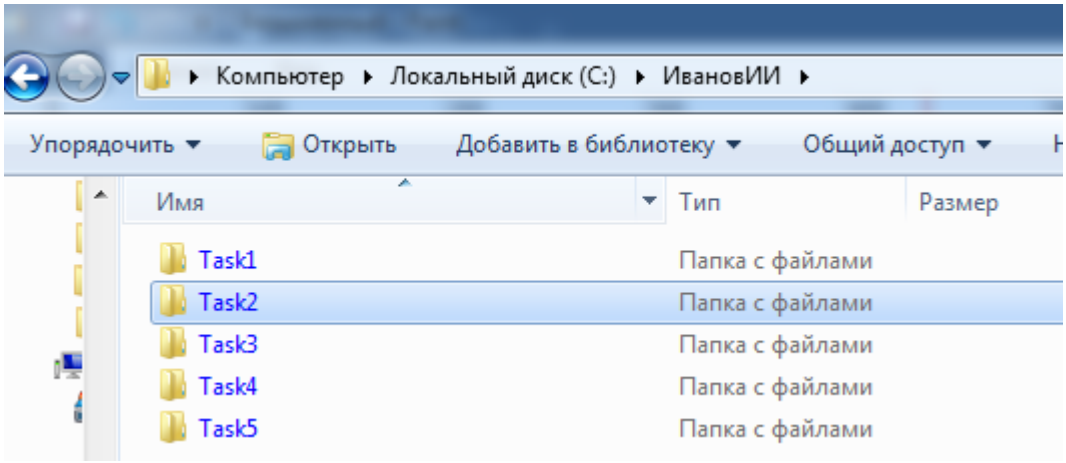


Рисунок П.1 – Структура каталога участника олимпиады

Участник олимпиады должен предоставить членам комиссии на проверку только файлы с исходными текстами программ, которые должны быть названы участником олимпиады в соответствии с выполняемым заданием, например, для языка Python: Task1.py.

Расширение файла должно соответствовать языку. Переименуйте файлы перед сдачей работы, если это необходимо. (см пример на рис. П.2)

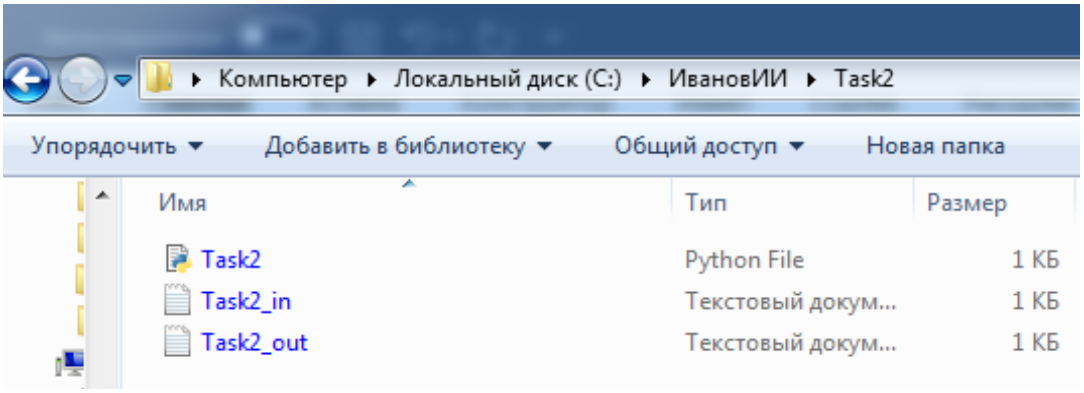


Рисунок П.2 – Размещение файлов в рамках папки задачи

В начале каждой программы должен находиться комментарий с ФИО участника, вариант, номером задачи, языком программирования, средой программирования или

онлайн-компилятора. Например, для С-подобных языков: // Иванов И. И., вариант 1, задача 1, Python 3.7.3, Spyder 3.3.6.

Если файлы с решением задачи, исходных и результирующих данных имеют некорректные названия и/или отсутствует первая строка комментарий, и/или размещены в каталоге участника без учета требований к структуре, то члены комиссии данное решение не оценивают и баллы за решение задания не начисляются.

При решении задач в качестве файлов с исходными данными и выходными данными используется только текстовый файл с расширением *.txt. Если в задаче программной реализации используется файлы с исходными данными и/или выходными данными, то кроме файла с исходным текстом требуется выслать соответствующие файлы. Например, для задания 2 требуется использовать исходные данные из файла, тогда название файла должно быть Task2_in.txt. если в задание 2 требуется сформировать текстовый файл с результатами исполнения программы, то название файла должно быть Task2_out.txt. Число после слова Task соответствует номеру решенного задания, «_in» определяет, что файл с исходными данными, «_out» определяет, что в файле хранятся результаты исполнения кода над исходными данными. Все текстовые файлы с исходными данными создаются участником самостоятельно, в соответствии с представленными в задачах примерами и шаблонами. Ввод и вывод текста осуществляется только латинскими буквами.

По окончанию работы над заданиями участник формирует архив с содержимым решений в соответствии с предложенной выше структурой. Расширение архивного файла должно быть rar или zip (пример см. рис П.3-П.5).

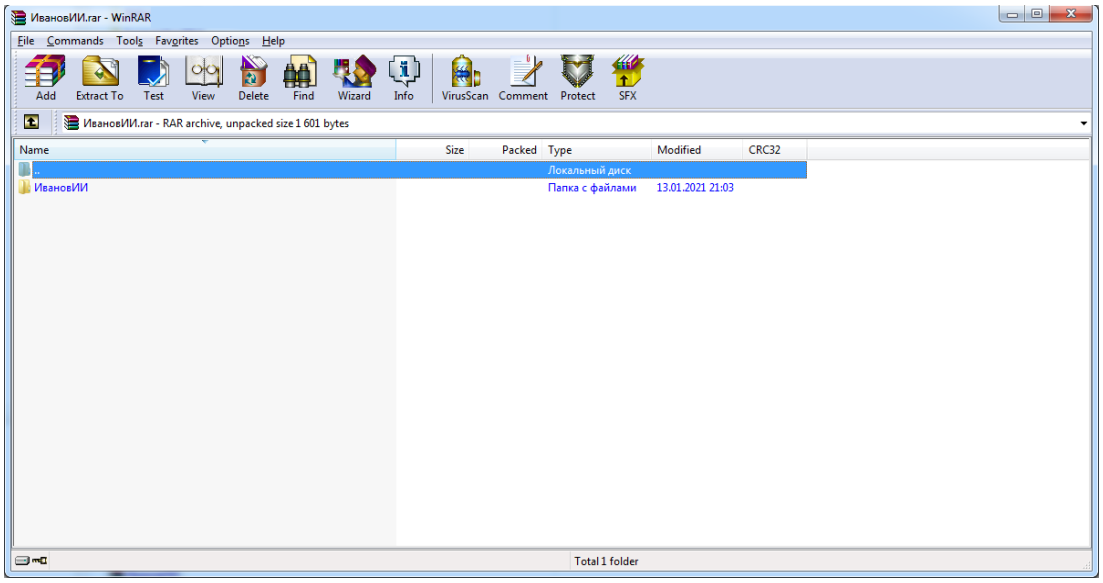


Рисунок П.3 – Пример первого уровня содержимого архива

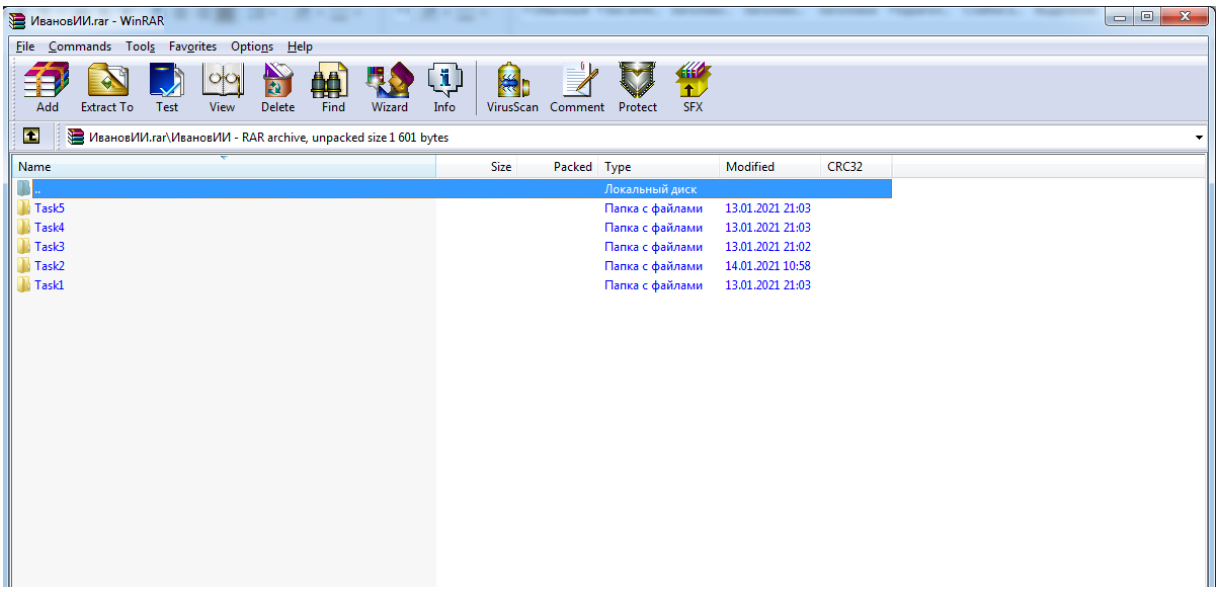


Рисунок П.3 – Пример второго уровня содержимого архива

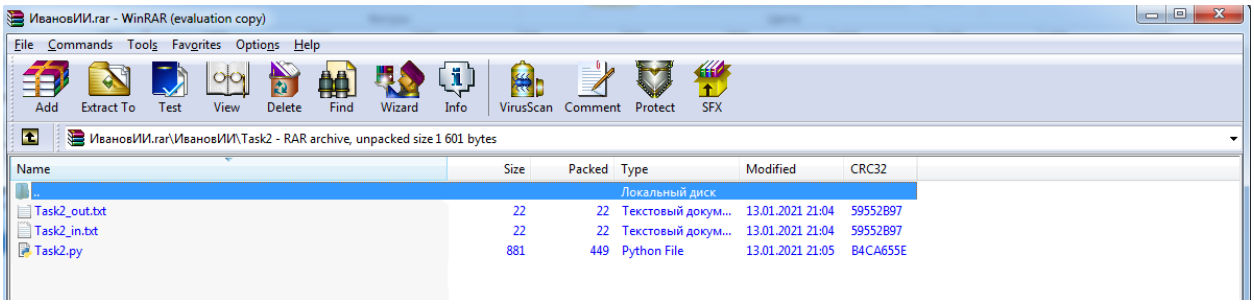


Рисунок П.3 – Пример третьего уровня содержимого архива

Члены комиссии, при проверке заданий, используют операционные системы семейства Windows (версии 7-10). Если члену комиссии не удается запустить файл с исходным кодом на исполнение (например, программа не выполняется в перечисленных ОС и/или не находит файл с исходными данными, и/или не формирует результирующий файл и т. д.), то задание считается не выполненным.

Максимальное количество баллов, которые может набрать участник – 100.

При оценивании решения задачи члены жюри могут снизить баллы за следующие недостатки:

- неполное соответствие решения условию;
- применение неэффективного алгоритма;
- решение задачи только для частного случая;
- отсутствие проверок, приводящих к снижению надежности программы;
- низкое качество интерфейса пользователя;
- несоответствие решения пулу тестовых значений;
- плохая читабельность текста программы и т. д.

При обнаружении использования участником: посторонней помощи в любом проявлении, средств интернет, мобильных устройств и других приемо-передающих устройств, способствующих решению заданий олимпиады, не используя собственные знания, приведут к исключению участника и аннулированию его результатов.

Для программной реализации заданий участник олимпиады должен использовать онлайн-компилятор https://www.onlinegdb.com/, пройдя процедуру регистрации.

При неработоспособности онлайн-компилятора https://www.onlinegdb.com/ участник олимпиады может использовать следующие среды разработки: CodeBlocks, Anaconda, Lazarus, Mono + MonoDevelop, установленные на ЭВМ.

В случае, если программный код участника не запускается в вышеперечисленных средах разработки, комиссия не рассматривает данную работу.

Задания на очный этап олимпиады «Информатика»
Вариант №1

Задание 1 (8 баллов)

Реализовать программу для расчета функции y=y(x), представленной на рисунке 1.1. Значения x вводятся с клавиатуры, результат вычислений выводится на экран ПК.

$$y = \frac{x}{x^2 - 1} + \log_3(x + 2)$$

Рисунок 1.1 — Функция y=y(x)

Задание 2 (12 баллов)

Выполнить программную реализацию расчета значений логической функции четырех переменных (см. рис. 2.1) и вывод результата в файл Task2_out.txt.

F = (A or C or D and not B) or (B or A ≡ D) and (D ← A)
Рисунок 2.1 – Логическая функция четырех переменных

В функции обозначения соответствуют:

- *or* – логическое ИЛИ (дизъюнкция);
- and – логическое И (конъюнкция);
- *not* – логическое отрицание (НЕ);
- ← – обратная импликация;
- ≡ – эквивалентность.

Исходные данные с 16 комбинациями значений переменных требуется считать из файла Task2_in.txt. (пример см. рис. 2.2, порядок столбцов A B C D всегда соответствует данному примеру и не может быть изменен).

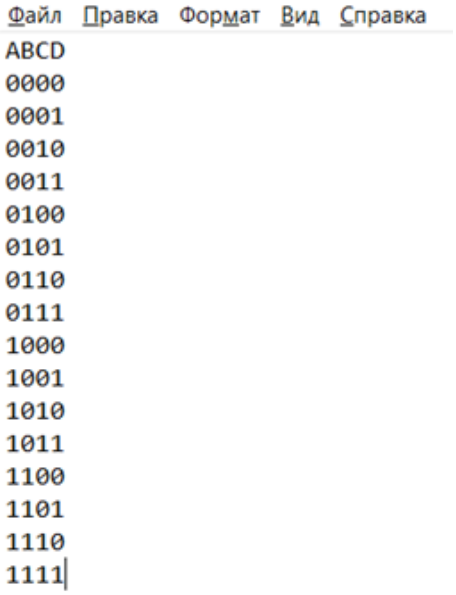
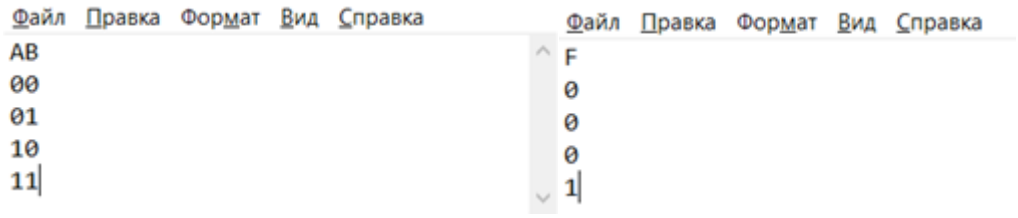


Рисунок 2.2 – Файл Task2_in.txt

Значение F вычисляется разработанной программой (за ручное решение с записью в файл Task2_out.txt баллы не начисляются), и записывается в файл Task2_out.txt.

На рисунках 2.3 a) и 2.3 b) показаны примеры файлов Task2_in.txt и Task2_out.txt, заполненных в соответствии с требованиями, для логической функции от двух переменных

F = A and B.



a) b)

Рисунок 2.3 – a) Файл Task2_in.txt, b) Файл Task2_out.txt

Задание 3 (20 баллов)

Александра и Антон решили использовать в своей переписке методы шифрования. Проанализировав длину сообщений, они выяснили, что она никогда не превышает 25 символов, включая знаки препинания и пробелы. На основе данного анализа ребята выбрали в качестве алгоритма шифрования ребята решили использовать одноалфавитный шифр Виженера.

Шифр Виженера – метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Для шифрования и расшифрования может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера, представленная на рисунке 3.1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Рисунок 3.1

Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

Пример работы данного алгоритма шифрования, следующий.

Допустим, что в качестве исходного алфавита используется:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Фраза, которую требуется зашифровать: HELLO.

Ключ для шифрования: KEY.

Человек, посылающий сообщение, записывает ключевое слово (KEY) циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: (KEYKE). Первый символ исходного текста H зашифрован последовательностью K, которая является первым символом ключа. Первый символ R шифрованного текста находится на пересечении строки K и столбца H в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста I получается на пересечении строки E и столбца E. Остальная часть исходного текста шифруется подобным способом.

Фраза для шифрования: HELLO.

Ключ для шифрования: KEYKE.

Зашифрованный текст: RIJVS.

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если буквы A-Z соответствуют числам 0-25, то шифрование Виженера можно записать в виде формулы:

$$c_i = (p_i + k_i)\ mod\ 26.$$

Расшифровка:

$$p_i = (c_i + 26 - k_i) mod\ 26,$$

где c_i– символ закодированного сообщения, p_i – символ исходного сообщения, k_i – символ ключа.

Помогите ребятам автоматизировать процесс шифрования и дешифрования с помощью описанного метода (пробелы в сообщениях не шифруются и соответственно не дешифруются).

Требования к программе:

1. Пользователь выбирает действие, которое собирается осуществить (реализация меню см. рис. 3.2):

```
Выберите действие:
Шифрование нажмите - 1
Дешифрование нажмите - 2
```

Рисунок 3.2

2. При нажатии «1» программное средство должно выполнить шифрование.

3. При нажатии «2» программное средство должно выполнить дешифрование.

При шифровании в качестве входного файла программа должна использовать файл Task3_in_SH.txt структура которого представлена на рисунке 3.3.

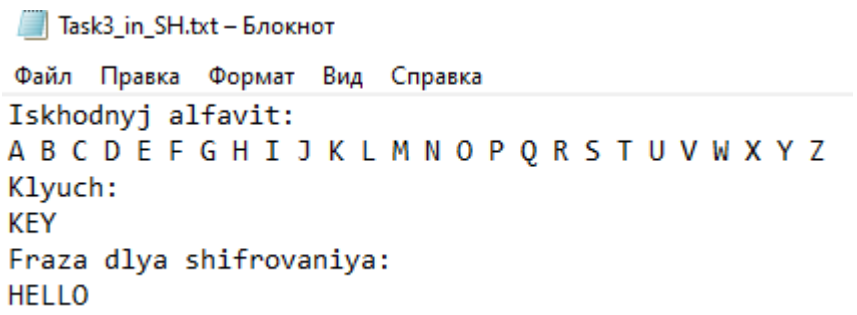


Рисунок 3.3

В качестве выходного файла формируется файл Task3_out_SH.txt структура которого представлена на рисунке 3.4.

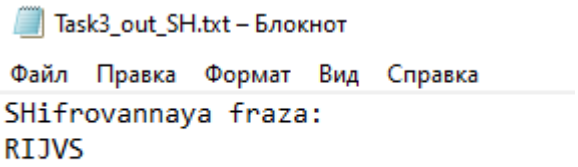


Рисунок 3.4

При дешифровании в качестве входного файла программа должна использовать файл Task3_in_DSH.txt структура которого представлена на рисунке 3.5.

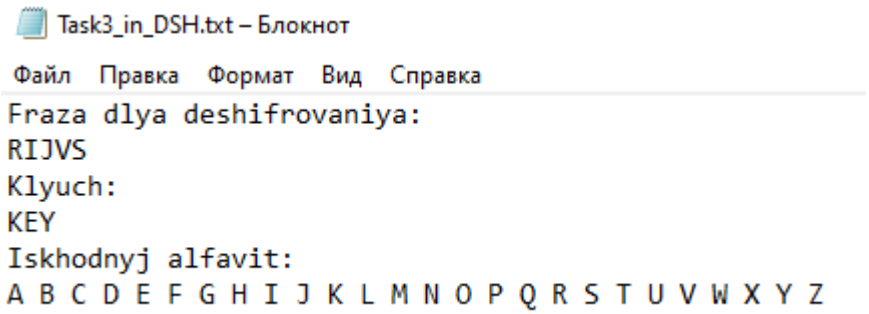


Рисунок 3.5

В качестве выходного файла при дешифровании формируется файл Task3_out_DSH.txt структура которого представлена на рисунке 3.6.

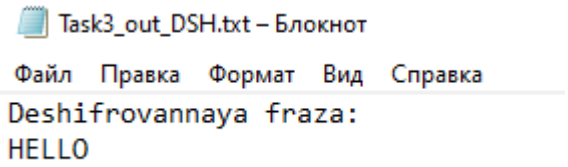


Рисунок 3.6

При шифровании и дешифровании фраз требуется выполнить проверки:

1. На существование в исходном алфавите всех символов, которые будут подвержены кодированию или дешифрованию (алфавит может включать как буквы, так и символы в соответствии с входным файлом);
2. Длина сообщения не превышает 25 символов, включая знаки препинания (пробелы из сообщения удаляются или сообщение вводится без пробелов).

При несоответствии требованиям программное средство должно сформировать следующие сообщения в Task3_out_SH:

1. «В исходном алфавите не указаны все символы, которые использованы в сообщении»;
2. «Длина сообщения не соответствует требованиям».

Задание 4 (25 баллов)

В Банке есть несколько филиалов в разных городах. Из филиала выезжает инкассаторская машина в центральный офис Банка, по дороге заезжая в разные города для смены водителя и охраны. Помогите сотрудникам банка автоматизировать поиск наикратчайшего пути.

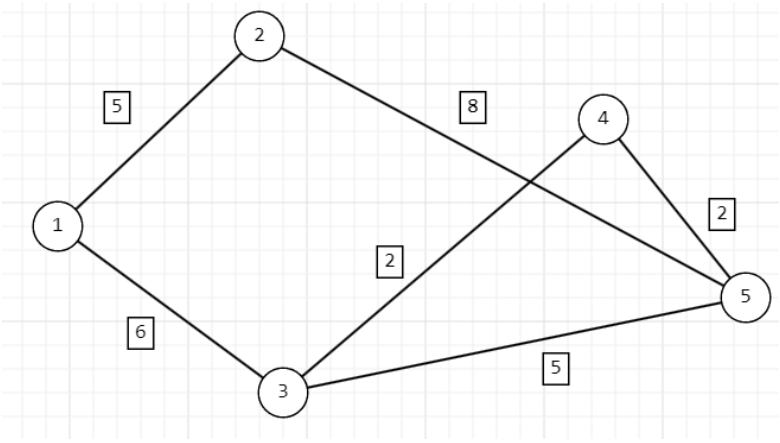


Рисунок 4.1

Пусть кружками обозначены города, а ребра расстояния между городами. В кружках обозначены номера вершин, над рёбрами обозначен их вес — длина пути. Необходимо построить наикратчайший путь от вершины 1 до вершины 5.

Для хранения весов графа используется квадратная матрица. В заголовках строк и столбцов находятся вершины графа. А веса дуг графа размещаются на пересечении вершин. Граф не содержит петель, поэтому на главной диагонали матрицы содержатся нулевые значения.

Т.е. для рисунка 1 получается матрица:

	V1	V2	V3	V4	V5
V1	0	5	6	0	0
V2	5	0	0	0	8
V3	6	2	0	0	5
V4	0	0	2	0	2
V5	0	8	2	2	0

0 5 6 0 0
5 0 0 0 8
6 2 0 0 5
0 0 2 0 2
0 8 2 2 0

В качестве входного файла программа должна использовать файл Task4_in.txt структура которого представлена на рисунке 4.2.

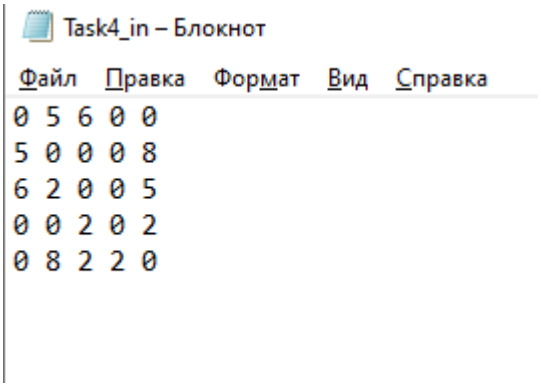


Рисунок 4.2

Вывод наикратчайшего пути: v1-v3-v4-v5

Полученные данные требуется сохранить в файл Task4_out.txt, структура которого показана на рисунке 4.3.

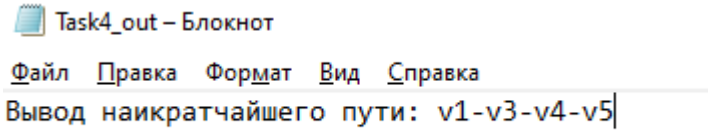


Рисунок 4.3

Задание 5 (35 баллов)

На шахматной доске находится черный король и белая фигура (слон, конь, ладья или ферзь). Требуется написать программу, которая определяет, может ли белая фигура дать шах черному королю при условии, что сейчас ход белых.

Описание

Обозначения:

Игровое поле для игры в шахматы представляет собой доску, разделенную на 64 квадратные клетки по 8 клеток по горизонтали и 8 клеток по вертикали. Каждая вертикаль обозначается латинской буквой от a до h, а каждая горизонталь - арабской цифрой от 1 до 8. Таким образом каждая клетка имеет свой уникальный адрес, например b2 или e4 (представлена на рисунке 5.1.).

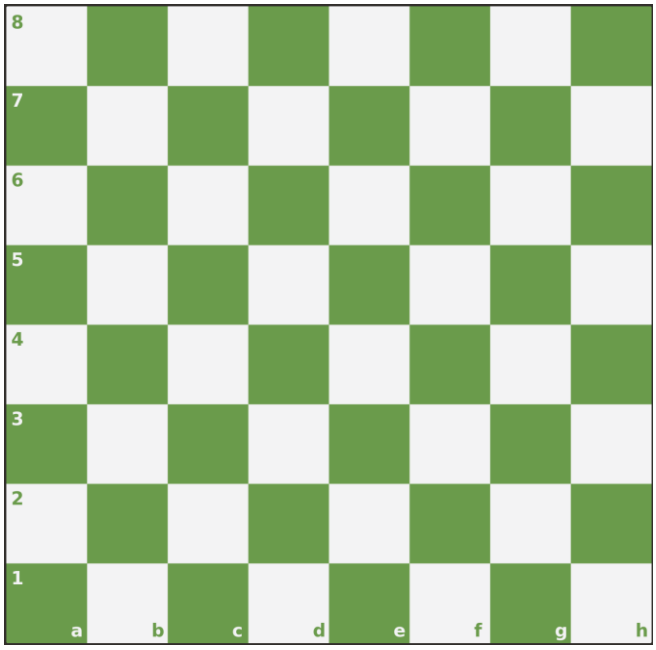


Рисунок 5.1

Шахматная фигура может находится на одной определенной клетке доски. На одной клетке может находится только одна фигура. Для записи положения фигур будут использоваться стандартное обозначение - имя фигуры и имя клетки, на которой располагаются фигуры. Фигуры обозначаются так: N - конь, B - слон, R - ладья, Q - ферзь, K - король. Пешка никак не обозначается. Так, например, запись Nb1 означает, что конь находится на клетке b1, запись e8 означает, что пешка находится на клетке e8. Пробел между именем фигуры и именем клетки не ставится.

Правила игры:

Пешки могут двигаться только на одну клетку по вертикали. Белые пешки двигаются “вверх” - в направлении увеличения числа в имени клетки, а черные - “вниз” - в

направлении уменьшения. Назад пешки ходить не могут. Исключение - белые пешки, находящиеся на 2 горизонтали, и черные - на 7 горизонтали. Эти пешки могут ходить как на одну клетку вперед, так и на две.

Конь движется на две клетки в одном из четырех направлений и на одну клетку в одном их двух, направлений, перпендикулярном первому:

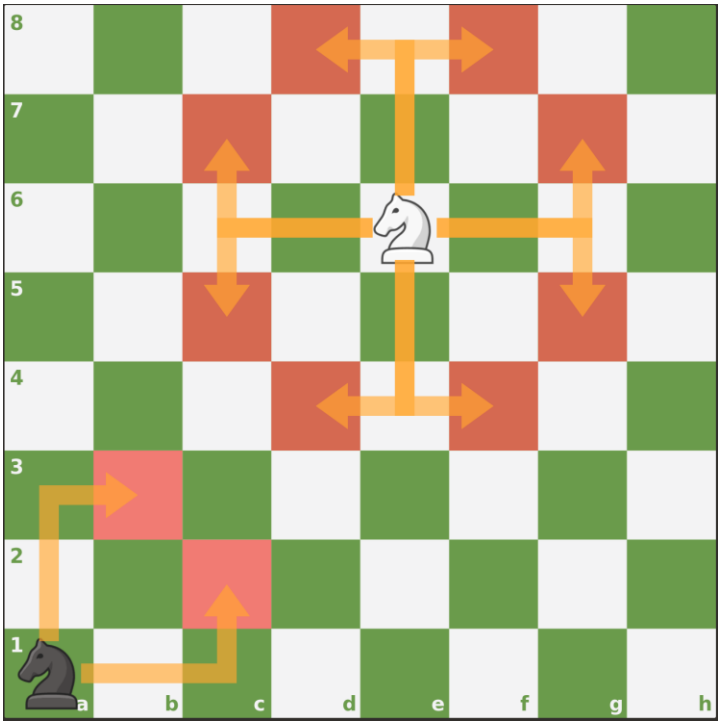


Рисунок 5.2

На рисунке 5.2 красным отмечены клетки, на которые может переместиться соответствующий конь. Конь, расположенный в середине доски может переместиться на любую из восьми клеток. Конь. расположенный в углу может переместиться на любую из двух клеток потому, что фигуры не могут выходить за пределы доски.

Ладья движется по горизонтали и вертикали на любое доступное количество клеток.

Слон движется по горизонтали на любое доступное количество клеток.

Фигуры не могут двигаться за пределы доски. Фигуры за исключением коня не могут “перепрыгивать” через другие фигуры, в том числе короля.

Шах - это ситуация, когда король находится на клетке под боем фигуры или пешки противоположного цвета. Клетка считается под боем фигуры тогда, когда эта фигура может следующим ходом переместиться со своего текущего положения на эту клетку. Исключение - пешки. Под боем пешки находятся клетки по диагонали:

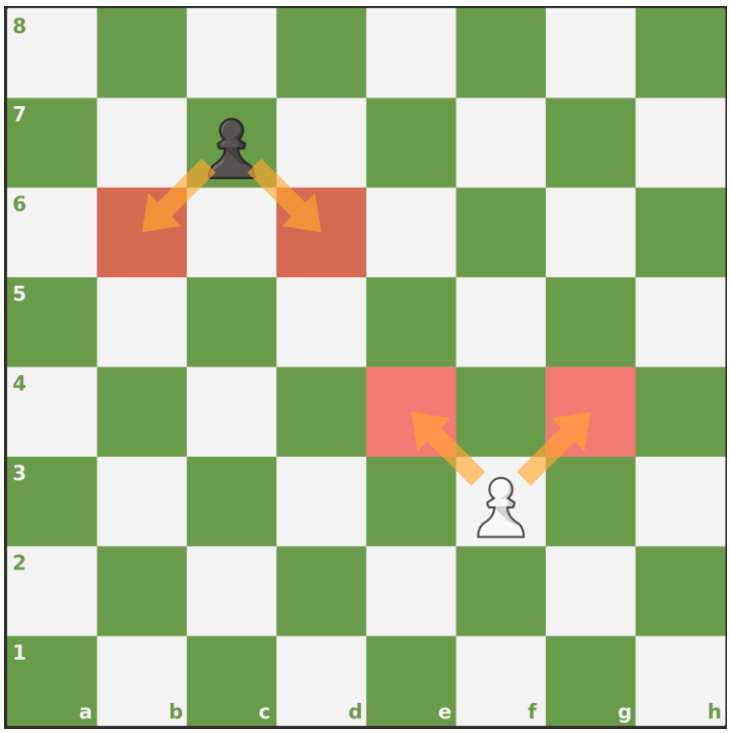


Рисунок 5.3

На рисунке 5.3 красным обозначены клетки, находящиеся под боем ближайших пешек. Обратите внимание, что направление боя для пешки зависит от ее цвета и, как следствие, направления движения.

Формат входного файла:

Во входном файле указаны положения белой фигуры и черного короля в одну строчку через пробел. Пример:

e4 Kf8

соответствует такому положению на доске (рисунок 5.4):

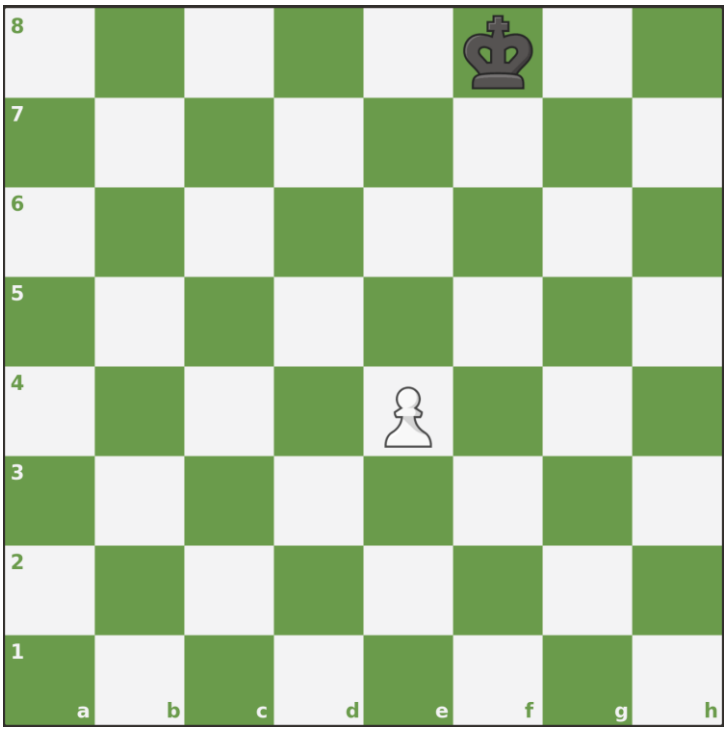


Рисунок 5.4

А такой файл:

e4 Kf8

соответствует такому положению на доске (рисунок 5.5):

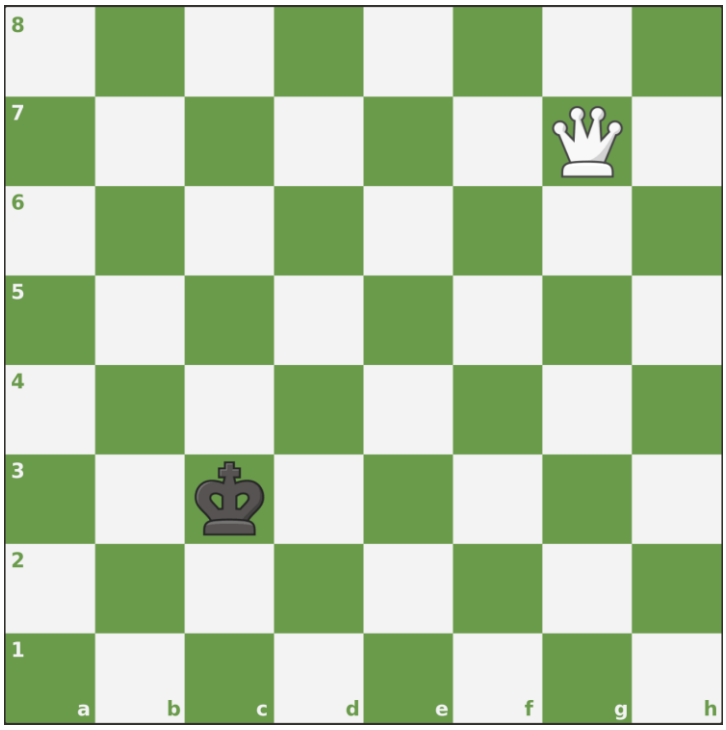


Рисунок 5.5

Формат выходного файла:

Если шах возможен, то программа должна записать в выходном файле соответствующих ход, то есть положение фигуры, на которое она должна переместиться для объявления шаха. Если таких ходов несколько, то программа должна вывести первый по алфавиту. Если черный король уже находится под шахом в исходном положении, программа должна вывести 0.

Если шах невозможен, то выходной файл должен быть пустым.

Пример: входной файл - “Be6 Ke2”. Выходной файл должен содержать “Bc4”.